



Selenium Open Source Test Automation Framework

Version 1.0

September 2009

DISCLAIMER

Verbatim copying and distribution of this entire article is permitted worldwide, without royalty, in any medium, provided this notice is preserved.

TABLE OF CONTENTS

1. PURPOSE OF THE DOCUMENT	3
1.1. Scope	3
1.2. Overview	3
2. FRAMEWORK CODE STRUCTURE	4
3. DRIVER FUNCTIONS	5
3.1. Keyword Driver Function	5
3.2. Main Function	5
4. ACTION FUNCTIONS	7
4.1. Perform Function	7
4.2. Store Function	8
4.3. Check Function	8
5. REPORTING AND ERROR-HANDLING FUNCTIONS	10
5.1. Reporting Function:	10
5.2. Error-Handling Function:	11
6. STRING EXPRESSION FUNCTIONS	12
6.1. Function for string operations:	12
7. COMMON FUNCTIONS	13
7.1. Function for Retrieving Variables:	13
7.2. Function for Press Key Operations:	13
7.3. Function for Dynamic Wait:	14
7.4. Function for Importing Keyword Script from Excel Sheet	14
8. REUSABLE FUNCTIONS	15
8.1. Function for Calling Reusable Actions	15
9. CONDITION AND LOOPING FUNCTIONS	16
9.1. Condition function	16
9.2. Loop Function	17
10. POPUP FUNCTION	18
10.1. Function for handling Windows popup:	18

1. Purpose of the Document

The purpose of this document is to describe the Open Source Test Automation Framework code in detail.

1.1. Scope

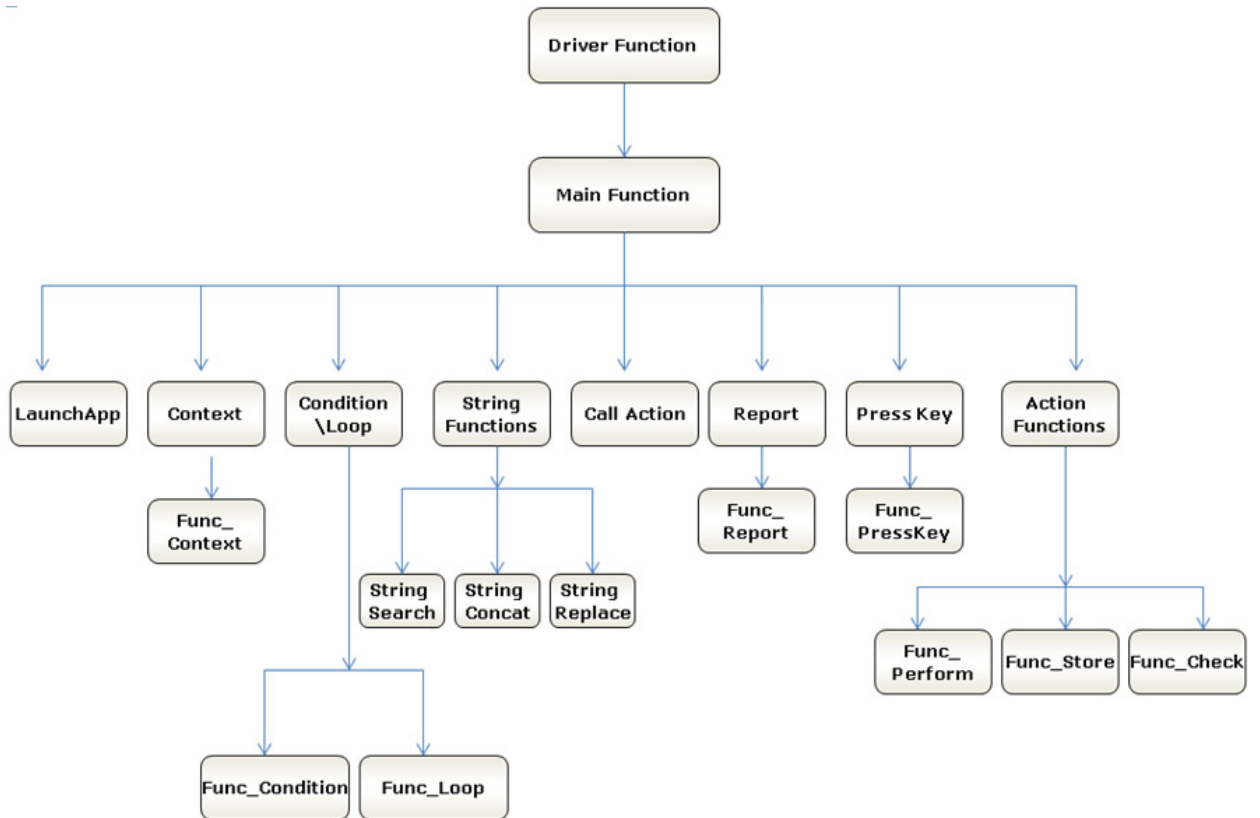
The scope of this document is to provide details about Open Source Test Automation Framework code and its architecture and functions.

1.2. Overview

This document provides details about:

- Framework Architecture
- Driver Functions
- Action Functions
- Reusable Functions
- Common Functions

2. Framework Code Structure



3. Driver Functions

3.1. Keyword Driver Function

Name of the function: Keyword_Driver ()

Description: This function is used to call the main framework.

Parameters: NA

Assumptions: The Automation Script is created and the Test Case Excel is ready.

Variables:

- a) intRowCount - Loops through all the data table rows
- b) strScriptName - Stores the name of the Script that has to be run

Functionality:

- Reads the values in the first column of the Test Case Excel.
- Whenever the value is 'r,' it calls the main function (Test_Step), which in turn will call (Keyword) function.
- When 'r' is not present in any of the cells in the first column, it skips the row and reads the value in the next row.

3.2. Main Function

Name of the function: Keyword ()

Description: This is the main function, which interprets the keywords and performs the desired actions. All the keywords used in the data table are processed in this function.

Parameters: NA

Variables:

- a) strAction - Stores the value in the second column
- b) cCellData - Stores the value in the third column
- c) dCellData - Stores the value in the fourth column

Functionality:

- Reads the values in the second, third, and fourth columns in the data sheet and stores the value in the variables. (Please refer to details in the Variables section.)
- Based on the value in the variable initial (2nd column), it calls different functions.
- If the value is something other than 'perform,' 'storevalue,' or 'check,' Keyword() calls the respective functions. For example, if the value is 'report,' it will call Report ().

Refer to the Framework code structure diagram to learn more about the function calls.

4. Action Functions

4.1. Perform Function

Name of the function: Perform ()

Description: This function performs the set of actions on the required object in AUT.

Parameters:

- a) CurObject - This refers to the object on which the specific operation needs to be performed.
- b) arrcCellData - This holds the type of the object on which the action has to be performed and the object name.
- c) arrdCellData - This is the operation that needs to be performed on the object.
- d) intRowCount - This holds the count of the current row in the data table.

Variables:

- a) cCellData[1] - Stores the values after splitting strParam with the delimiter ";"
- b) dCellData[1] - Stores the values after splitting strParam with the delimiter ":"
- c) CurObject - Stores the object on which the action has to be done
- d) col - Stores the number of columns in the table

Functionality:

- Based on the values in dCellData(0), Func_Perform () performs different actions. If the value is:
 - i) set: Checks the value in cCellData(0). If the value is 'textbox,' 'webfile,' 'tabletextbox,' 'checkbox,' or 'tablecheckbox,' then the value in dcelldata(1) is set.
 - ii) click: This operation is performed on the current object.
 - iii) selectindex: Checks for the value in cCellData(0). If the value is 'radiobutton,' 'combobox,' 'tablecombobox,' or 'tableradiobutton,' then the item with the index number in the arrKeyIndex(1) variable is selected in the object.
 - iv) select: Checks for the value in cCellData(0). If the value is 'radiobutton,' 'combobox,' 'tablecombobox,' or 'tableradiobutton,' then the item with the value in the arrKeyIndex(1) variable is selected in the object.
 - v) close: Checks for the value in the variable cCellData(0). If the value is "browser," "window," or "dialog," then it will perform a close operation on the parent object.
 - vi) submit: Performs a submit operation on the current object. The object must be in focus for this operation.
 - vii) deselect: Checks for the variable value in cCellData(0). If the value is "combobox," then the deselect operations of the value in the variable arrKeyIndex(1) are performed on the current object.

4.2. Store Function

Name of the function: storevalue ()

Description: This function is used to store any property of a particular object into a variable.

Parameters:

- a) Object - This is an object on which the specified operation needs to be performed.
- b) Value - This holds the type of value that need to be stored in an object.

Assumptions: Context is set on the current object where the action has to be performed.

Variables:

- a) cCellData - Holds the object information
- b) dCellData - Holds the actions to be performed on the object

Functionality:

- Based on the values in cCellData(0), it performs different actions.
 - i) If the value in cCellData(0) is specified as 'textbox' then it stores the value present in the text box. The same is applicable for combo box also.
 - ii) Store value function can also be used to store the value in a variable if the object exists. For more details on the Keyword, please refer Selenium_framework_keywords.

4.3. Check Function

Name of the function: Check()

Description: This function is used for all the checking operations to be performed on the AUT.

Parameters:

- a) Object - This is the object on which the specified operation needs to be performed.
- b) cCellData - This holds the type of the object and the object name on which the check operation has to be performed.
- c) dCellData - This is the operation that needs to be performed on the object.
- d) intRowCount - This holds the count of the current row in the data table.

Variables:

cCellData - Stores the object on which the check condition need to be executed

dCellData - Stores the action that needs to be checked in the object

Functionality:

- Based on the values in dCellData(0), it performs different actions. If the value is:

- i) enabled: Checks if the specified object is enabled.
 - ii) visible: Checks if the specified object is Visible.
 - iii) focused: Checks if the specified object is Focused.
 - iv) selection: Checks if the Combo box has the value selected that is specified in dCellData(1).
 - v) checked: Checks if the Check box or Radio button is checked or selected.
 - vi) exist: Checks if the page or a browser with the specified title exists.
- If the values match, then it will generate a pass report. Otherwise, it will generate a fail report.

5. Reporting and Error-handling Functions

5.1. Reporting Function:

Name of the function: Report ()

Description: This function is used for generating a customized report with specified user inputs through the use of keywords.

Parameters: None

Assumptions: NA

Variables:

- a) reportobj - Stores the contents of the third column of the current row in the Global Sheet
- b) reportcon - Stores the status of the report (Pass/Fail)
- c) reportcon1 - Stores the actual message of the report
- d) reporter0 - Stores the expected message of the report
- e) expmess - Stores the concatenated expected message
- f) actmess - Stores the concatenated actual message

Functionality:

- 'reportobj' is split with delimiter ";" and is stored in the array 'reportcon.'
- 'reportcon(0)' holds the status of the report.
- 'reportcon(1)' is split with delimiter ":" and is stored in the array 'reportconl.'
- 'reportconl(0)' is split with the delimiter ":" and is stored in the 'reporter0.'
- 'reporter0' holds the expected message.
- 'reportconl(1)' is split with the delimiter ":" and is stored in the 'reporter1.'
- 'reporter1' holds the actual message.
- Based on the values in the 'reportcon(0),' the Func_Report () function will generate different reports. If the value is :
 - i) Pass:
 - Generates a report with status as Pass, expected message as 'reporter0,' and actual message as 'reporter1'
 - ii) Fail:
 - Generates a report with status as Fail, expected message as 'reporter0,' and actual message as 'reporter1'
 - iii) Done:
 - Generates a report with status as Done, expected message as 'reporter0,' and actual message as 'reporter1'

iv) Warning:

- Generates a report with status as Warning, expected message as 'reporter0,' and actual message as 'reporter1'

5.2. Error-Handling Function:

Name of the function: Recovery()

Description: This function is used to capture the error generated at runtime.

Parameters: NA

Assumptions: NA

Variables:

a) intRowCount - This holds the row count of the step that is executed currently.

Functionality:

If there is an error in the execution, then this function will be called. This function by default will skip the current line of execution and will proceed. However, users can customize the function to their requirements to continue the next line, to end the test suite execution, or to bring the application to base state.

6. String Expression Functions

6.1. Function for string operations:

Name of the function: Func_StringOperations ()

Description: This function is used for all string operations.

Parameters:

- a) strCriteria - This variable holds the value of the second column of the data table.

Assumptions: None

Variables:

- a) arrSplit - Stores the elements from the third column of the data table after splitting with the ";" delimiter
- b) strMainString - Stores the main string (arrSplit(0))
- c) strSubString - Stores the sub string(arrSplit(1))
- d) intLen - Stores the length of the array "arrSplit"
- e) ReturnVal - Stores the return value

Functionality:

- Based on the values in strCriteria, Func_StringOperations () performs different actions. If the value is:
 - i) strsearch:
 - 1. Searches for the substring (strSubString) in the main string (strMainString)
 - 2. Stores the position of the substring in the return value variable (ReturnVal)
 - ii) strconcat:
 - 1. Concatenates the main string (strMainString) and the substring (strSubString)
 - 2. Stores the concatenated string in the return value variable (ReturnVal)
 - iii) strreplace:
 - 1. Searches for the substring (strSubString) in the main string (strMainString) and replaces it with strString (arrSplit(2))
 - 2. Stores the replaced main string in the return value variable (ReturnVal)
- After the ReturnVal variable is updated, the value in the ReturnVal variable is stored in the variable specified in the fourth column of the data table.

7. Common Functions

7.1. Function for Retrieving Variables:

Name of the function: GetValue ()

Description: This function is used to retrieve the value from any variable.

Parameters:

a) strCellData - This variable holds the type of the object and the object name on which action should be performed or checked.

Assumptions: NA

Variables:

a) arrSplitCheckData - Stores the row number in the table where the text is present

b) strParamName - Stores the flag to check if text is found or not

Functionality:

- This function searches for '#' in the variable strCellData. If '#' is present then the environment value of the value in the variable is returned by the function.
- If '#' is not present, then the variable strCellData is split with the delimiter "_" and is stored in the array arrSplitCheckData.
- Based on the values in the variable arrSplitCheckData(0), the GetValue () function performs different actions. If the value is:
 - i) p: Parameter value of the variable in arrSplitCheckData(0) is returned by the function
 - ii) dtl: Retrieves the value from the data sheet (second worksheet) available along with the script excel sheet
 - iii) dt: Retrieves the value from the cell in action 1 sheet with the column name in the variable arrSplitCheckData(0) and row in the variable intDataCounter

7.2. Function for Press Key Operations:

Name of the function: Func_presskey ()

Description: This function is used to retrieve the value from any variable.

Parameters:

a) arrObj - This variable holds the value of the third column in the data table.

b) WshShell - This object was created for shell scripting.

Assumptions: NA

Variables: NA

Functionality:

- WshShell object is created.

- Based on the values in arrObj(0), different values are passed as arguments to the SendKeys method of the created shell scripting object.
- Shell object is set to 'nothing.'

7.3. Function for Dynamic Wait:

Name of the function: Wait ()

Description: This function is used for synchronization with the application.

Parameters:

- a) arrObj - This variable holds the value of the third column in the data table.
- b) arrKeyValue - This is the operation that needs to be performed on the object.

Assumptions: NA

Variables: NA

Functionality:

- A function check for the upper boundary of the array arrKeyValue is greater than or equal to zero or not.
- If yes, then the Func_Wait () function will check for the value of the variable arrKeyValue(0).
- If the value is "exist" then the Func_Wait () function will call the WaitProperty method of the object 'currentparent' by passing the arguments "visible," True, and 10000.

7.4. Function for Importing Keyword Script from Excel Sheet

Name of the function: Func_ImportData ()

Description: This function is used to import test data at runtime.

Parameters:

- a) strTestCase - This variable holds the file name and the path where it is stored.

Assumptions: The required file is present and it is an Excel sheet.

Variables:

- a) strPath - Stores the Path with .xls
- b) strDataPath - Locates and stores the full Path
- c) strSheetName - Stores the sheet name to be imported

Functionality:

- This function will store the value of the fourth column in the variable 'strSheetName.'
- Using the ImportSheet method, the Excel sheet with the name in the variable 'strSheetName' in the Excel work book in the path specified by the variable 'strDataPath' is imported to the Action1 sheet.

8. Reusable Functions

8.1. Function for Calling Reusable Actions

Name of the function: Func_CallAction ()

Description: This function is used to call a reusable Action.

Parameters:

- a) strData - Holds the name of the reusable action.
- b) strInfo - Holds the parameters for the reusable action.

Assumptions: NA

Variables:

- a) arrParam - Stores the parameters to be passed to the reusable action
- b) strActionName - Stores the reusable action name

Functionality:

- This function checks for the value in the variable strInfo.
- If the value is Null, then the Func_CallAction () function will call the RunAction method by passing variable 'actionName' and value 'oneIteration' as arguments (without passing parameters).
- If the value is not Null, then the Func_CallAction () function will split the value in the variable strInfo with the delimiter ':' and store it in array 'paramSplit.'
- Based on the number of items in array paramSplit, this function performs different actions.
- For example, if the number of items in paramSplit is 4, then 4 parameters are passed while calling the RunAction method.

9. Condition and Looping Functions

9.1. Condition function

Name of the function: Func_Condition ()

Description: This function is used to evaluate the expression according to the inputs given in keyword script.

Parameters:

a) intRowCount - Holds the value of current row number of the data table

Assumptions: NA

Variables:

a) iFlag - Sets the flag

b) cndSplit - Stores the value of the fourth column of the Global Sheet

c) startRow - Stores the start row for the condition

d) endRow - Stores the end row for the condition

e) cstrCellData - Stores the condition to be checked

f) var1 - Stores the first element to be evaluated

g) var2 - Stores the second element to be evaluated

Functionality:

- cndSplit array is stored with values in the fourth column of the data table after splitting with the delimiter “;”.
- The variable startRow is assigned with the value of the first item in the array cndSplit.
- The variable endRow is assigned with the value of the second item in the array cndSplit.
- cstrCellData array is stored with the values in the third column of the data table after splitting with the delimiter “;”.
- The variable var1 is assigned with the value of the first item in the array cstrCellData.
- The variable var2 is assigned with the value of the third item in the array cstrCellData.
- Then the condition in the cstrCellData(1) is evaluated and the intRowCount is assigned with the value in the startRow if the condition is true. Otherwise, intRowCount is assigned with the value in the endRow if the condition is false.

9.2. Loop Function

Name of the function: Func_loop ()

Description: This function is used to repeat a set of statements for a specified number of times.

Parameters:

- a) variable - Holds the query variable that has to be converted
- b) strconverttype - Holds the data type into which the variable is to be converted

Assumptions: If the number of times to be looped is not specified, by default this number is taken as the number of active rows in the Action1 sheet of the data table.

Variables:

- a) arrloopData - Stores the start row and end row values
- b) intcntr - Stores the loop count
- c) Counter - Stores the count value
- d) endRow1 - Stores the end row for looping
- e) loopRowCount - Stores the current loop count
- f) intDataCounter - Stores the current data counter

Functionality:

- arrloopData is stored with the values after splitting the value in the third column of the data table with “;” as a delimiter.
- The variable intcntr is assigned with value in the fourth column in the data table.
- Value in the variable intcntr is converted into an integer and is stored in the variable Counter.
- The value in the variable intcntr is stored in the variable Counter.
- This function recursively calls Keyword_Web function (Main function) ‘n’ times. Here, ‘n’ is the value present in the variable Counter.

10. Popup Function

10.1. Function for handling Windows popup:

Name of the function: Popup ()

Description: This function is used to handle Windows popup

Parameters:

- a) Thread Name -Initialize the new thread and perform the action
- b) Autoit - Initialize the variable to access autoit commands

Assumptions: AutoIt has been installed in the system.

Variables:

- a) cCellData - Stores array after the value has been split out using ; as delimiter
- b) dCellData - Stores array after the value has been split out using : as delimiter

Functionality:

- If a windows popup comes as part of clicking a button or launching a URL, then that action will be executed in a separate thread.
- Enters the value in a popup textbox.
- Clicks on the button in a popup.
- Stores the inner text in the popup.

COPYRIGHT

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.